Cross-Layer Attack on Anonymizing Networks

Andriy Panchenko
Department of Computer Science IV
RWTH Aachen University
Ahornstr. 55, D-52074 Aachen
Email: panchenko@cs.rwth-aachen.de

Lexi Pimenidis
Department of Computer Science IV
RWTH Aachen University
Ahornstr. 55, D-52074 Aachen
Email: pimenidis@nets.rwth-aachen.de

***Abstract*— Network layer anonymization protects only some of the user's personal identification information, namely network addresses of the communicating parties. However, even if the lower layers of communication provide perfect protection for the user's profile, information leakage on the application layer destroys the whole effort. Currently, all widespread implementations of anonymizing networks do not use a holistic approach and therefore, neither filter nor actively warn users about information leakage from the upper layers, which may look innocent to the end user.**

We extend existing work on security of anonymizing networks to take into account additional information leakage from the application layer. Further we show, under which conditions and how this kind of information can be used not only to build an extensive user profile at "low costs", but also to speed up traditional attacks that are targeted at the network layer identification of users' peer partners.

I. Introduction

Whereas cryptography has the goal of content protection of the messages, anonymous communication deals with the relationship hiding between the communicating parties. Providing such a protection in an open environment is a very challenging task that is getting more and more attention. In today's digitalized world, demand for this is emerging not only from the industry and commerce, but also from private individuals. Many approaches have been proposed to protect privacy on the Internet. Still, only a few of them have been implemented in praxis, e.g. [1], [2].

The general idea for anonymizing data traffic is to send messages not directly, but through several so called "middle nodes". This way, the relaying nodes gain no knowledge whether the relayed data streams are just redirected on behalf of the others, or actually originate from the predecessor node. It is usually distinguished between *high-latency* and *low-latency* anonymization systems. Former are used for non time critical traffic, like e.g. e-mail or usenet and often rely on Chaum's mix principle [3]. The latter ones are designed for real-time communication, like e.g. web-browsing or instant messaging.

There is a number of proposals and practical implementations of anonymization networks (see e.g. [4]). Most of them are based on mixing [3], onion routing [1], randomized routing protocols [5], or on DC-networks [6]. A number of attacks exist, especially on the low latency implementations (c.f. [7]) that are not trivial to defend. Those, based on the DC-networks, can be used to provide perfect anonymity under some assumptions [6], however the protocol has serious drawbacks causing questionable practical implementation, i.e. it requires secure and reliable broadcast channels, is prone to channel jamming, inefficient in large networks, etc. [4].

All the widespread solutions provide protection solely on the network layer. Without this protection an attacker can get information about the IP address of the involved sender and recipient, time, duration, and volume of the communication. Over time, any of this can possibly be enough to uniquely determine a person's identity.

Due to the massive deployment of anonymization networks in recent time, it has become possible to check their properties "in the wild". As it can be seen from the recent research, the most successful attacks were not targeted to the network layer, but rather either at the application layer [8] or at the side channels [9]. This shows evidently that an increasing number of serious attacks target parts of the network stack that were not subject of extensive research in the past. Moreover, since the anonymization techniques are not working holistically and network protocols on the application layer were not designed originally in a privacy-friendly way, a possibility to get additional information about the users on application layer (e.g. in HTTP headers there are cookies, accepted languages tags, browsers and OS version, etc.) exists for the attackers.

Thus, application layer profiles can be enreached with the information from the network layer, and vice versa. If an attacker gets information on any layer of the communication system, he can use this information in order to reduce the anonymity on the other layers as well. However, in this case an attacker has to make sure that the same entities are observed on both the layers.

It seems ridiculous to assume that attackers restrict their analysis to only a single communication layer and will not make use of all the information available to them. Therefore, calculations on a user's degree of anonymity should not blend out the application layer, even if the published information seems innocent at the first glance.

A. Contribution

Our contribution in this area is the following:

1) We show a model for information leakages on application layer, especially in the area of anonymous web browsing.

978-1-4244-2036-0/08/$25.00

2) A method of linking these data is proposed. There we also show, to which extent attacking application layer is related to known network layer attacks.
3) We present a calculation to estimate the accuracy of linking these data items as well as speed up for known network layer attacks using the information from the application layer.

B. Roadmap

This paper is structured as follows: in Section II we list a set of papers that cover previous work on this topic and discuss them. Sections III and IV contain the main contribution, namely the model and all calculations. The paper ends up with conclusions in Section V.

II. Related Works

There are, to the best of our knowledge, only a few works on the anonymity analysis that take into account data on the application layer. The lack of attacks' evaluation that consider cross-layer information is, however, a serious drawback in the security evaluation of anonymization techniques. Considering only a single layer gives a wrong perception about the anonymity properties of the considered protocols: harmless settings from one layer perspective may mean complete deanonymization when a holistic approach is taken for mounting the attack. And it is naive to assume than an attacker will not make use of it.

Irwin and Yu [10] argue that because of the inferences, the privacy of users may still be compromised even if powerful credential and anonymous communication systems are used. The authors pinpoint the need for dynamically adjustable privacy policies for users' protection in open environments. These policies depend on the identifiability and information sensitivity of the data. An influence of the former depends on the size of the anonymity set, while the latter depends on the appraisal of the user itself. It is, however, possible that a set of data that seems to be innocent to the user can be used in order to deanonymize him.

Similar results are shown in [11], [12], where despite the usage of strong cryptographic elements, like e.g. zero knowledge proofs, users of the system can be profiled and identified.

Clauß and Schiffner in their work on anonymity on application layer [13] present a statistical attacker model and propose the use of it for user anonymity measures when publishing personal data. A user that wants to publish an attribute set can estimate how this information can be used in order to deanonymize him. Any subset of attributes is called either partial identity or profile. They distinguish these two terms in the following way: "partial identity" is used if it is known that the origin of the set of attributes is a single entity. A set of attributes without having an evident binding to an entity is called "profile". However, this work is rather theoretical and too abstract in order to be used as part of an actual attack.

In their other work on structuring anonymity metrics [14], they present models for anonymity metrics for both – network and application layer. Further, the need for merging those models (from both layers) is raised in order to provide usable metric for user-centric identity management system. Still, the practical realization of the combined model and metrics is left out for the future work.

Finally, Diaz et. al. [15] shows a counterexample, where the anonymity (measured in terms of entropy) increases when the adversary combines information from the network and application layers. In contrast to this, in our contribution we provide a necessary condition for the decrease of anonymity. Having this condition helps the users to estimate to which extent their behavior contributes to the speed up of their deanonymization.

III. Background & Model

Application layer data can be used not only to create extensive users' profiles, but also, in addition to the information gained on the network layer, to facilitate and to speed up the deanonymization process of classical attacks on the network layer. By this term, we mean identification of the communication relations between peers in an anonymization network.

It should not be misunderstood, that an application layer profile has to be unique and must contain personal identifying information (PII) in order to mount a successful deanonymization. The case where obvious PII is presented on the application layer is trivial and will not be discussed here. The point is, that even innocent looking information on the application layer (e.g. browser version, operation system, etc.) helps to speedup traditional attacks on the network layer drastically. Even if filtering on the application layer is performed and possibly unique tags are substituted by more general ones, the attack is still successful as long as there are entities with different communication profiles.

We will use the predecessor attack [16] as an example in this work to show how a network layer attack and simultaneous application layer profiling can benefit from each other. The choice was done in order to have algorithms which allow a thorough mathematical analysis; however it is highly probable that other instances of profiling and network layer attacks can also be combined. The approach presented below can be applied for different anonymization protocols, e.g. for onion routing [1], Crowds [5], ANTs [17], or mixing [3].

In this section we first come up with a background on the predecessor attack and then define a model for a generic anonymization system to show in the follow up section, how application layer information can be used to speed up the predecessor attack on the network layer and create an extensive profile of the user.

A. Background

Low-latency anonymization techniques are known to be vulnerable to the predecessor attack [16], [18], [19], [20], [21]. In the predecessor attack, one or more attackers are members of the anonymization network and observe the communication of their path predecessors during a number of rounds (path

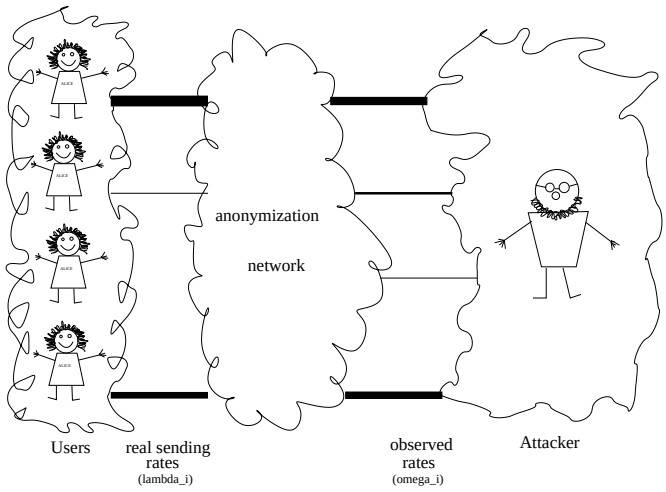

Fig. 1. Information Leakage

or session reformations). It was mentioned above, that the anonymity is reached by means of an increased path length: sending a message not directly, but through some number of intermediate nodes. Doing so the relation between a sender and a receiver is being hidden. But whenever an attacker is able to determine some specific session (thus, the tunnel goes through at least one node belonging to the attacker), there is some first attacker that sees the message. Only the first occurrence of the attacker in a path is interesting, since it is the closest position to the victim – the message originator.

Assumed that long-lived interactions (over extended period of time) between senders and receivers exist, after a certain number of rounds a particular sender will appear to the attacker as the initiator of the communication with some non-negligible probability. This is based on the fact [16], [21], [22], that the path predecessor is more likely to be originator (we call it probability p_h) than any other participant (let's call this probability p_l)[1] of the network. Therefore, $p_h > p_l$.

Thus, there exist an information flow in the system: an attacker statistically sees more messages from those users that have higher communication rates, i.e. send messages on their own and not only forward them on behalf of the others. The interdependency between the real communication rate and the rate observed by the attacker is depicted in Figure 1. The thickness of the line corresponds to the sending rate of the corresponding user (the thicker it is, the higher is the sending rate). Note, that the missing line from one of the users on the left-hand side means that he/she does not send own messages, but rather only forwards some on behalf of the others (which corresponds to the thin line on the right-hand side).

Now, we describe how the canonical predecessor attack as presented in [19], [16], [20] can be applied to create an extensive profile also on the application layer and this, in return, can be used for speeding-up the canonical predecessor attack on the network layer. Thus, making it faster and more dangerous than used to think before. But first we continue with the model.

[1] h stands for "high", l for "low"

B. Model

Without loss of generality, we will restrict our analysis to a single application layer property only in this section. For an extension to multiple properties, check the discussion in Section IV-B.

Let $\mathcal{N} = \{A_1, A_2, \ldots A_n\}$ be the set of all the $n = |\mathcal{N}|$ users in the anonymizing system. Further, let $\mathcal{M} = \{L_1, L_2, \ldots L_m\}$ be the set of all possible values of some attribute in the communication profile (e.g. accepted languages in HTTP request). Similarly, $m = |\mathcal{M}|$ is the number of possible values of the attribute.

We basically distinguish the following three events:

$Src(A_i)$ - A_i is the real sender of the message;
$Rx(A_i)$ – message was received by attacker from A_i;
$MSG(L_j)$ – message has attribute value L_j.

Again, without loss of generality, we are interested in finding out the communication rates of all honest users with some third party[2]. Frequency consideration is important because sending a few messages to some third party is not necessary a sign for practically having it as a confirmed communication party. For example, consider a user who receives a link per e-mail and is tricked to click on it, or is prone to a XSS-attack[3], which downloads images from an arbitrary Internet host. This could lead to a false positive categorization which can be filtered out by considering rates.

W.l.o.g. lets assume that in an arbitrary fixed time interval, each user A_i sends messages generated on its own to the anonymization network with the rate λ_i (this rate can also be equal zero). An attacker is a member of the network and controls at least one node. On the long run there will be cases where A_i is a direct path predecessor of one of the nodes controlled by the attacker. As a direct path successor he observes the rate ω_i of messages coming from A_i. In low-latency anonymization networks, there is a correlation between real own sending rate λ_i and the rate ω_i observed by an attacker which serves as a basis for an attack [16], [21], [22].

Depending on the applied anonymization technique, different requirements arise for the position of the attacker in the network: first node on the path in case of Crowds [5], or first one and the last one in case of onion routing [23], etc. This issue is addressed in detail in [16], [21], [22]. In this work we just assume that an attacker is in the position that allows him to mount the predecessor attack.

We extend the model as follows: let $\Pi(A_i, L_j)$ denote the actual rate of messages with attribute value L_j (an example of an attribute is e.g. excepted languages field of the HTTP request) that are originated by A_i. Similarly, $\Phi(A_i, L_j)$ is the sending rate of messages as observed by the attacker with attribute value L_j that are received from A_i. We use $\tilde{\Phi}(A_i, L_j)$ for the estimated value by the attacker for the $\Pi(A_i, L_j)$.

[2] The algorithm can be trivially extended to identify rates to all receivers and scales linearly.

[3] Cross-site scripting: http://www.cert.org/advisories/CA-2000-02.html

Next, we provide the calculation for cross-layer deanonymization that makes use of the introduced model. The information from both – network and application layers is combined in order to speed up the classical attack.

IV. Calculation

In this section we deduce formulas to quantify the information gain for an attacker, that take into account application layer data. The central variable in this area is p_h, which denotes the following: given that an attacker is on the path, this is the probability that the direct predecessor of the first attacker node in a message's path is the original sender of the message [20].

We are looking for p_h^{new}, the probability that A_i is the originator of the message, given that message is received from A_i by an attacker and that it has an attribute L_j:

$$p_h^{new} = p(Src(A_i)|Rx(A_i), MSG(L_j)) \tag{1}$$

Using the rule of the conditional probability two times and reforming, we get:

$$p_h^{new} = \frac{p(Rx(A_i)|Src(A_i), MSG(L_j))p(Src(A_i), MSG(L_j))}{p(Rx(A_i), MSG(L_j))} \tag{2}$$

Because forwarding in anonymization networks does not depend on the content of a message, the following holds: $p(Rx(A_i)|Src(A_i), MSG(L_j)) = p(Rx(A_i)|Src(A_i))$. Furthermore, applying Bayes to all three components of Equation 2 and taking into account, that

$$p(Src(A_i)|Rx(A_i)) = p_h, \tag{3}$$

$$p(MSG(L_j)|Src(A_i)) \underset{t\to\infty}{=} \frac{\tilde{\Phi}(A_i, L_j)}{\sum_{q=1}^{m} \tilde{\Phi}(A_i, L_q)}, \tag{4}$$

$$p(MSG(L_j)|Rx(A_i)) = \frac{\Phi(A_i, L_j)}{\sum_{q=1}^{m} \Phi(A_i, L_q)}, \tag{5}$$

where t is the number of observations (please note that also the follow up formulas are valid only for $t \to \infty$), the following holds:

$$p_h^{new} = \frac{p_h \frac{p(Rx(A_i))}{p(Src(A_i))} \quad p(Src(A_i)) \frac{\tilde{\Phi}(A_i, L_j)}{\sum_{q=1}^{m} \tilde{\Phi}(A_i, L_q)}}{p(Rx(A_i)) \quad \frac{\Phi(A_i, L_j)}{\sum_{q=1}^{m} \Phi(A_i, L_q)}} \tag{6}$$

$$= p_h \cdot \frac{\tilde{\Phi}(A_i, L_j) \quad \sum_{q=1}^{m} \Phi(A_i, L_q)}{\Phi(A_i, L_j) \quad \sum_{q=1}^{m} \tilde{\Phi}(A_i, L_q)} \tag{7}$$

$$= p_h \cdot \frac{p(Msg(L_j)|Src(A_i))}{p(Msg(L_j)|Rx(A_i))}. \tag{8}$$

As already mentioned above, from the observations of $\Phi(A_i, L_j)$ it is possible to draw conclusions about the $\tilde{\Phi}(A_i, L_j)$ (for $t \to \infty$):

$$(\Phi(A_1, L_j), \ldots, \Phi(A_n, L_j))^T = M(\Pi(A_1, L_j), \ldots, \Pi(A_n, L_j))^T \tag{9}$$

$$\iff (\tilde{\Phi}(A_1, L_j), \ldots, \tilde{\Phi}(A_n, L_j))^T = M^{-1}(\Phi(A_1, L_j), \ldots, \Phi(A_n, L_j))^T \tag{10}$$

where the matrix M is as follows:

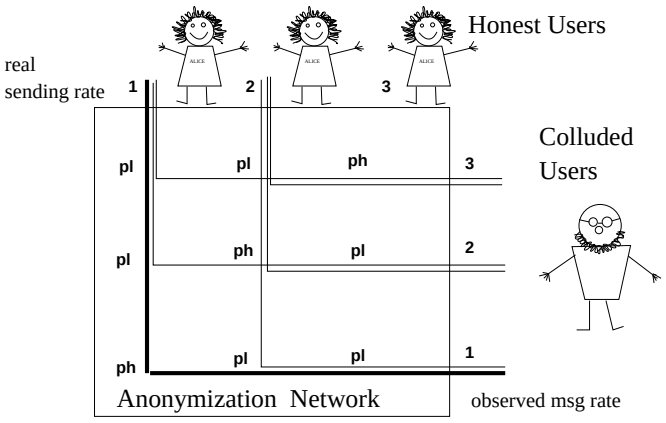

Fig. 2. Matrix A

$$M = \begin{pmatrix} p_h & p_l & \cdots & p_l \\ p_l & p_h & \cdots & p_l \\ \vdots & \vdots & & \vdots \\ p_l & p_l & \cdots & p_h \end{pmatrix} \tag{11}$$

Indeed, the matrix M is invertible to M^{-1} because low-latency anonymization networks are unable to provide perfect security ($p_h \neq p_l$) [19], [20], thus:

$$M^{-1} = \begin{pmatrix} \tilde{p_h} & \tilde{p_l} & \cdots & \tilde{p_l} \\ \tilde{p_l} & \tilde{p_h} & \cdots & \tilde{p_l} \\ \vdots & \vdots & & \vdots \\ \tilde{p_l} & \tilde{p_l} & \cdots & \tilde{p_h} \end{pmatrix} \tag{12}$$

where

$$\tilde{p_h} = \frac{n+p_h-2}{np_h-1} \tag{13}$$

$$\tilde{p_l} = \frac{p_h-1}{np_h-1} \tag{14}$$

How the matrix M is associated to the real system is depicted in Figure 2. The number of messages that are observed by an attacker from an honest user is the user's own sending rate times probability p_h plus the sum of all the messages generated by the other users times probability p_l. In Figure 2, the third user merely forwards messages on behalf of the others, without inserting its own traffic. Still, the attacker as a direct path successor observes some amount of messages coming from the third user.

Thus, we have:

$$\tilde{\Phi}(A_i, L_j) = \tilde{p_h} \cdot \Phi(A_i, L_j) + \tilde{p_l} \cdot \sum_{k \neq i} \Phi(A_k, L_j) \tag{15}$$

Let's also define the following sums for convenience purposes

$$\sum_{i=1}^{n} \Phi(A_i, L_j) \quad := \quad \Phi_{\Sigma L_j} \tag{16}$$

$$\sum_{q=1}^{m} \Phi(A_i, L_q) \quad := \quad \Phi_{\Sigma A_i} \tag{17}$$

$$\sum_{i=1}^{n} \sum_{q=1}^{m} \Phi(A_i, L_q) \quad := \quad \Phi_{\Sigma} \tag{18}$$

From here it follows that

$$\frac{p(Msg(L_j)|Src(A_i))}{p(Msg(L_j)|Rx(A_i))} = \frac{\tilde{\Phi}(A_i, L_j) \cdot \sum_{q=1}^{m} \Phi(A_i, L_q)}{\Phi(A_i, L_j) \cdot \sum_{q=1}^{m} \tilde{\Phi}(A_i, L_q)} \underset{t \to \infty}{=} \tag{19}$$

$$\frac{\Phi(A_i, L_j) \cdot \Phi_{\Sigma A_i} \cdot (\tilde{p}_h - \tilde{p}_l) + \tilde{p}_l \cdot \Phi_{\Sigma L_j} \cdot \Phi_{\Sigma A_i}}{\Phi(A_i, L_j) \cdot \Phi_{\Sigma A_i} \cdot (\tilde{p}_h - \tilde{p}_l) + \tilde{p}_l \cdot \Phi(A_i, L_j) \cdot \Phi_{\Sigma}} \tag{20}$$

Furthermore, since

$$\tilde{p}_h - \tilde{p}_l \quad = \quad \frac{n-1}{np_h - 1} \tag{21}$$

we come to the final equation:

$$\frac{\Phi(A_i, L_j) \cdot \Phi_{\Sigma A_i} \cdot (\tilde{p}_h - \tilde{p}_l) + \tilde{p}_l \cdot \Phi_{\Sigma L_j} \cdot \Phi_{\Sigma A_i}}{\Phi(A_i, L_j) \cdot \Phi_{\Sigma A_i} \cdot (\tilde{p}_h - \tilde{p}_l) + \tilde{p}_l \cdot \Phi(A_i, L_j) \cdot \Phi_{\Sigma}} \quad = \tag{22}$$

$$\frac{(n-1) \cdot \Phi(A_i, L_j) \cdot \Phi_{\Sigma A_i} + (p_h - 1) \cdot \Phi_{\Sigma L_j} \cdot \Phi_{\Sigma A_i}}{(n-1) \cdot \Phi(A_i, L_j) \cdot \Phi_{\Sigma A_i} + (p_h - 1) \cdot \Phi(A_i, L_j) \cdot \Phi_{\Sigma}} \tag{23}$$

Thus,

$$p_h^{new} = p_h \cdot \frac{\Phi_{\Sigma A_i} \cdot ((n-1) \cdot \Phi(A_i, L_j) + (p_h - 1) \cdot \Phi_{\Sigma L_j})}{\Phi(A_i, L_j) \cdot ((n-1) \cdot \Phi_{\Sigma A_i} + (p_h - 1) \cdot \Phi_{\Sigma})} \tag{24}$$

The latter will be used to show the information gain taking into account the information on the application layer. Please note that so far we have considered only a single property on the application layer.

A. Information gain

From Formula 8 it follows that an attacker gains information, iff

$$\frac{p(Msg(L_j)|Src(A_i))}{p(Msg(L_j)|Rx(A_i))} > 1 \tag{25}$$

$$\text{(see 23)} \quad \Leftrightarrow \quad \Phi_{\Sigma L_j} \cdot \Phi_{\Sigma A_i} < \Phi(A_i, L_j) \cdot \Phi_{\Sigma} \tag{26}$$

$$\Leftrightarrow \quad \frac{\Phi_{\Sigma A_i}}{\Phi_{\Sigma}} < \frac{\Phi(A_i, L_j)}{\Phi_{\Sigma L_j}} \tag{27}$$

Therefore, an attacker can gain additional information when the frequency of messages with a certain property observed from a user, relative to all messages that are observed from the anonymization network with that property, is higher than the quote of all the messages observed from the user in the whole anonymization network.

Please note, that the observed message rate $\Phi(A_i, L_j)$ correlates with the real sending rate $\Pi(A_i, L_j)$. Therefore, it is possible to say that an attacker can gain information, *if the part of the user's messages with attribute L_j in the system is higher than the user's quote of all messages in the system*. E.g. there are 10 users in a group. Each sends a message in each round. Six of the users send messages in English. Thus, an attacker will be able to gain additional information from the system using application layer even with respect to the users' communication in English. It should also be noted, that those that send messages in other less popular languages will be even more identifiable due to the higher p_h^{new} value.

B. Network Layer Attack Speedup

The classical predecessor attack aims to identify a user's peer partners at the network layer only. We will show in this section that finding a user's peer partners by a predecessor attack can be sped up by building an extensive user profile on the application layer in parallel, i.e. identifying values of different communication attributes.

While a user typically communicates with many arbitrary peers, his application layer profile (set of accepted languages, browser version, etc.) remains usually the same. Therefore, an attacker will discover the application layer profile of his victim much faster than the communication profile. If the attacker is using a statistical attack on the network layer, he can then use information from the application layer to bias the input for the network layer attack, e.g. by filtering out improbable combinations or in general, messages that do not fit to the victim's profile.

Recall that the required number of observations to confirm (identify) with arbitrary precision the user A_i's peer partner B proportionally depends on the rate of A_i to B observed by colluded users [19]. Therefore, by mounting a predecessor attack on application layer attributes, we are k times faster in identifying the user's attributes values rather than peer partners, assumed that the quotient of communication to B of user A_i is only $1/k$-th of his overall communication. This attack can of course be done in parallel for m different attributes, in order to build an extensive application layer profile of the user. This will result in a profile, whose values can be estimated with arbitrary precision, similarly to the classical predecessor attack [16], [21], [22], [19] on the network layer.

After the profiles of users A_i are built, it is possible to use the same data in order to identify the user's peer partners. Actually, even during the process of building the application layer profile, the calculations for the network layer can already be adapted on the fly. The attack works as follows:

- from the whole pool of observations the number of messages is extracted, received from each node having application layer attribute value L_j. This is stored in a vertical vector $\Phi(L_j)$
- for this selected attribute value, for each A_i an own $p_h^{new}(A_i)$ is calculated according to Formula 24.
- similarly to $p_h^{new}(A_i)$, $p_l^{new}(A_i)$ is derived:

$$p_l^{new} = p(Src(A_i)|Rx(A_k), MSG(L_j)) = \tag{28}$$

$$p_l \cdot \frac{\Phi_{\Sigma A_k} \cdot ((n-1) \cdot \Phi(A_k, L_j) + (p_h - 1) \cdot \Phi_{\Sigma L_j})}{\Phi(A_k, L_j) \cdot ((n-1) \cdot \Phi_{\Sigma A_i} + (p_h - 1) \cdot \Phi_{\Sigma})} \tag{29}$$

- matrix M_{L_j} is built, where each $M_{L_j}[i,i] = p_h^{new}(A_i)$ and $M_{L_j}[i, k \neq i] = p_l^{new}(A_i)$.

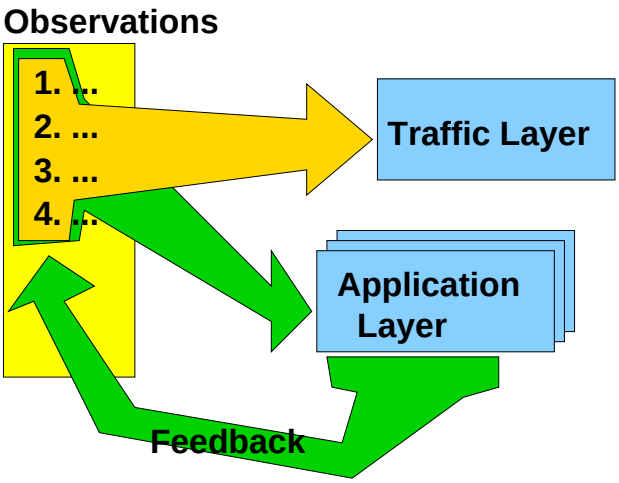

Fig. 3. Information Flow

- matrix M_{L_j} is inverted to $M_{L_j}^{-1}$.
- product of matrix $M_{L_j}^{-1}$ with the vector $\Phi(L_j)$ gives, as stated in Formula 10, vector of estimated user sending rates $\tilde{\Phi}(L_j)$
- the procedure is repeated for all L_j
- the result $\tilde{\Phi}_{final}$ is the vector with elements $\tilde{\Phi}_{final}[A_i] = \sum_{j=1}^{m} \tilde{\Phi}(L_j)[A_i]$

Figure 3 illustrates how the attack works: at first a user's application layer profile is built applying classical predecessor attack - but rather on application layer data. This information is further used in order to refine the classical attack on the network layer: the feedback is given to improve the observations which are used as an input to the attack on the traffic layer.

This combined attack is faster since the user profile is usually much more stable than just the communication profile. A quantification of the speedup is, however, beyond the scope of this paper and therefore subject to future work. The same holds for the practical realization and confirmation of the attack in the real-world settings.

The attack can be circumvented if the filtering system on the application layer would substitute the identifying information from the browser with the known statistical distribution of the data, like e.g. [24]. To be even more unidentifiable, one could observe the communication of the others as an attacker does, and calculate this statistic on his own. This is due to the reason that distribution of the identifying information among the users using anonymizing system may be different from those on the Internet in general. Another possibility would be to reduce sending rates of communication with some specific profile, so that no additional information leakage exist. But for the latter still the passive network observation mentioned above has to be performed.

V. Conclusions

Traditional attacks on anonymity networks are focused solely on the network layer. However, due to their generic nature we have shown how at least one of them, namely the predecessor attack, can be applied to data transmitted on the application layer too. Carried out on both layers, the attack is more precise and possibly faster. We have shown a condition under which application layer data leaks additional information for a network layer attack and proposed a strategy how to overcome this information flow.

We have also indicated, that it is insufficient to estimate the success of an attack by using a single layer only, as it has been done so far. Even seemingly innocent information (e.g. browser strings in HTTP requests) can be used in order to speed up traditional attacks on the network layer. It seems naive to think that an attacker would not make use of all the information which is available. Information from both network and application layer has to be considered in future research in order to provide usable and realistic metrics for anonymity.

This also shows the need for holistic approaches to anonymity, since even perfect protection on the network layer alone does not fulfill its goal if information is leaked on the other layers.

References

[1] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," in *Proceedings of the 13th USENIX Security Symposium*, 2004.

[2] O. Berthold, H. Federrath, and S. Köpsell, "Web MIXes: A system for anonymous and unobservable Internet access," in *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, H. Federrath, Ed. Springer-Verlag, LNCS 2009, July 2000, pp. 115–129.

[3] D. L. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84 – 88, Feb 1981.

[4] J.-F. Raymond, "Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems," in *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, H. Federrath, Ed. Springer-Verlag, LNCS 2009, July 2000, pp. 10–29.

[5] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for Web Transactions," *ACM Transactions on Information and System Security*, pp. 66 – 92, April 1998.

[6] D. L. Chaum, "The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability," *Journal of Cryptology*, no. 1, pp. 65 – 75, 1988.

[7] S. J. Murdoch and G. Danezis, "Low-Cost Traffic Analysis of Tor," in *Proceedings of the 2005 IEEE Symposium on Security and Privacy*. Oakland, California, USA: IEEE Computer Society Press, May 2005.

[8] FortConsult, "Practical onion hacking: Finding the real address of tor clients," October 2006. [Online]. Available: http://packetstormsecurity.nl/0610-advisories/Practical_Onion_Hacking.pdf

[9] S. J. Murdoch, "Hot or not: Revealing hidden services by their clock skew," in *13th ACM Conference on Computer and Communications Security (ACM CCS)*, Alexandria, Virginia, USA, 2006.

[10] K. Irwin and T. Yu, "An identifiability-based access control model for privacy protection in open systems," in *WPES '04: Proceedings of the 2004 ACM workshop on Privacy in the electronic society*. New York, NY, USA: ACM Press, 2004, pp. 43–43.

[11] A. Pashalidis and B. Meyer, "Linking Anonymous Transactions: The Consistent View Attack," in *Proceedings of the Sixth Workshop on Privacy Enhancing Technologies (PET 2006)*, G. Danezis and P. Golle, Eds. Cambridge, UK: Springer, June 2006.

[12] D. Kesdogan, V. Pham, and L. Pimenidis, "Information disclosure in identity management," in *Proceedings of 12th Nordic Workshop on IT-Security, NordSec*, Reykjavik, Iceland, Oct 2007.

[13] S. Clauß and S. Schiffner, "Anonymität auf Anwendungsebene," in *Proceedings of Sicherheit 2006*. Bonn, Germany: GI Lecture Notes in Informatics P-77, 2006, pp. 171–182.

[14] ——, "Structuring anonymity metrics," in *DIM '06: Proceedings of the second ACM workshop on Digital identity management*. New York, NY, USA: ACM Press, 2006, pp. 55–62.
[15] C. Diaz, C. Troncoso, and G. Danezis, "Does additional information always reduce anonymity?" in *WPES '07: Proceedings of the 2007 ACM workshop on Privacy in Electronic Society*. New York, NY, USA: ACM, 2007, pp. 72–75.
[16] M. Wright, M. Adler, B. N. Levine, and C. Shields, "The predecessor attack: An analysis of a threat to anonymous communications systems," in *ACM Transactions on Information and System Security TISSEC'04*, vol. 7 (4). ACM Press, November 2004, pp. 489 – 522.
[17] "ANTS File Sharing," http://antsp2p.sourceforge.net/, 2007, visited Aug 2007.
[18] V. Shmatikov, "Probabilistic model checking of an anonymity system," *Journal of Computer Security*, vol. 12, no. 3-4, pp. 355–377, 2004.
[19] A. Panchenko and L. Pimenidis, "Fundamental Limits of Anonymity Provided by Crowds," in *8th International Symposium on System and Information Security SSI'2006*, Sao Paulo, Brazil, 8-10 November 2006.
[20] ——, "Crowds revisited: Practically effective predecessor attack," in *Proceedings of the 12th Nordic Workshop on Secure IT-Systems (NordSec 2007)*, Reykjavik, Iceland, October 2007.
[21] M. Wright, M. Adler, B. N. Levine, and C. Shields, "An analysis of the degradation of anonymous protocols," in *Proceedings of the Network and Distributed Security Symposium - NDSS '02*. IEEE, February 2002.
[22] ——, "Defending anonymous communication against passive logging attacks," in *Proceedings of the 2003 IEEE Symposium on Security and Privacy*. IEEE, May 2003.
[23] P. F. Syverson, D. M. Goldschlag, and M. G. Reed, "Anonymous connections and onion routing," in *Proceedings of the IEEE Symposium on Security and Privacy*. Oakland, California, USA: IEEE Computer Society, 1997, pp. 44–54.
[24] "Browser statistics," http://www.w3schools.com/browsers/browsers_stats.asp, April 2008.